

A SITUATED AGENT-BASED DESIGN ASSISTANT

JOHN S GERO AND WEI PENG

Key Centre of Design Computing and Cognition

University of Sydney, Australia

Email address: {john, wpeng}@arch.usyd.edu.au}

Abstract. This paper introduces a situated agent-based design assistant. The agent is wrapped around an existing design tool in order to adapt that design tool to its use. Current design tools are unchanged by their use and as a consequence as the designer develops experience in using the tool, the tool remains the same. Such an agent is able to learn from its interactions with environments through its “experience”. This learning is based on situated representation mechanisms and a constructive memory system. The agent wrapper will be able to adapt the tool’s behaviours to its use and as a result improve the tool’s usability.

1. Introduction

Designing is a process that produces the structure of a designed artifact in fulfilling expected function and behaviour within interactions between designer and the environment. Designing can be regarded as sequences of situated acts in which a designer changes the trajectory of the developing design (Gero, 1998). Designers reflect in their designing activities (Schon, 1983), in which their understandings of the designs at hand change with interactions with the design environment based on their reinterpreting current design problems or situations from their experiences. What can be cognized (sensed, perceived, or conceptualized) by a designer is not predefined but rather constructed during interaction with the developing design, biased by the designer’s experiences and expectations. The design process therefore cannot be planned and predicted in advance. In order to assist designers in this dynamic process, we need tools that can deal with the interactions within a dynamic environment. However, current design tools, contrary to designers who gain experiences as they are designing, are unchanged by their use (Gero, 1996; Gero, 2003) and repeat themselves irrespective of their interaction with the design environment. It is of obvious

benefit that design tools maintain objective knowledge that is independent of its application, so as to be used with arbitrary problems (Gero, 1996). Therefore, how can we make a design tool that adapts to its use in order to improve the efficiency and effectiveness of further usage, while still maintaining its objective knowledge? We aim to address this issue by developing a situated agent assistant that is wrapped around an existing design tool and learns from its interactions with the design environment that includes the designer, the design and other agents, Figure 1. Through the agency provided, the tool should be able to gain experience and adapt its behaviour to its use, and as a consequence improve its usability without impairing its objective knowledge.

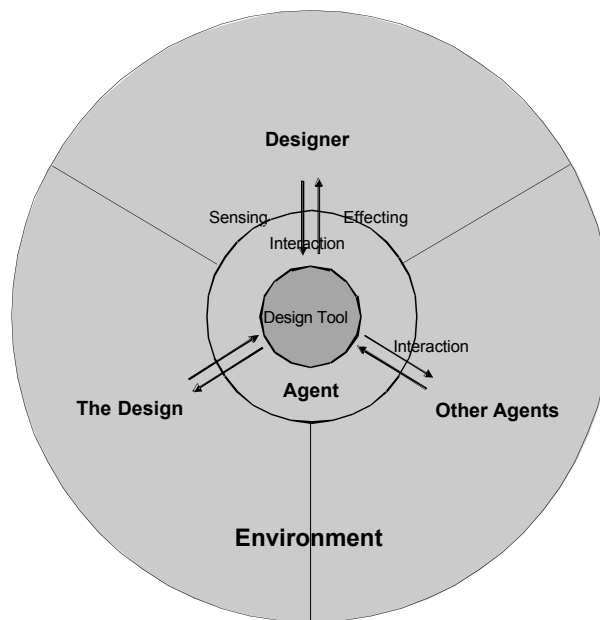


Figure 1. A situated agent assistant as a wrapper to a design tool.

2. Design Tool as a Situated Agent-Based Design Assistant

2.1. OTHER APPROACHES

Software agents are intentional systems that work autonomously and interact with environments in selecting actions to achieve goals (Wooldridge and Jennings, 1995). Traditional approaches fall into three groups. The first

group is the rule-based agent whose actions are pre-programmed and triggered by environment events, eg, the Oval system (Malone et al., 1987) acts (create, move and delete mails) based on specified rules. But this approach fails to deal with changes in the environment that cannot be predefined. The second group that involves endowing agents with extensive knowledge and is built on expert systems but is unable to adapt their fixed knowledge to the dynamic environments. The third group involves PBE (program by example) and PBD (program by demonstration) that were developed to allow a software agent to record the interactions between the user and a conventional “direct-manipulation” application. The agent then writes a program or script to respond to the user’s actions (Lieberman, 2001). How to generalize the program to accommodate changing situations is still the major problem for PBE (Lieberman, 2001). Although PBE uses artificial intelligence and machine learning techniques to address uncertainty, the adaptation it manifests has little impact on the agent’s experiences.

2.2. DESIGN TOOL AS A SITUATED AGENT-BASED ASSISTANT

We describe a design tool that can adapt its behaviour to it use to assist designers in utilizing that tool, through being wrapped by a situated agent-based design assistant which founds its adaptation on “situatedness” (Clancey, 1997; Gero, 1998b) and a constructive memory model (Gero, 1998a; 1999). Adaptation, the agent’s ability to accommodate incremental changes in the environment, enables the design tool to learn from and cope with the interactions in the dynamic design process. The interactions with designers, design representation and knowledge of other agents are constructed, grounded and reinforced into experiences. These experiences bias later memory construction when a similar situation is next constructed. The constructive memory model embodies a mechanism whereby an agent learns. Through the notion of a wrapper, the agency can be attached to the tool without changing its objective knowledge that is independent of its application (Gero, 2003).

2.2.1. *Situatedness*

Situatedness is the notion of “where you are when you do what you do matters” (Gero, 1998b). It states that an agent’s knowledge depends on the context in which it situated. What can be cognized is also related to agent’s experiences which are grounded from memory constructed through agent-environment interactions.

2.2.2. *Constructive Memory and Experience Grounding*

Constructive memory challenges the current static view of memory as a place to hold things. Memory is regarded as a learning process in which “memories are constructed initially from that experience in response to demands for a memory of that experience but the construction of the memory includes the situation pertaining at the time of the demand for the memory” (Gero, 1999). In this paper, we claim that memories are constructed from the experiential response to active situational cues. It is biased by current environment cues, the activated memory, and past experiences. An agent constructs memory from the current situation based on matching environment cues with experiences.

What if the current situation does not match any experiences? The agent needs processes to deal with changes of environment within its ongoing interactions with the environment. We view memory construction as being inseparable from the agent’s representation processes. Memory is constructed through the agent’s internal representation processes triggered both from external “data-driven” and goal-related “expectation-driven” demands (Gero and Fujii, 2000). It is constructed at a particular time, location, goal, internal and outward relationships, etc. Memory construction is biased by experience, goal (in terms of affecting expectation) and the influence of this constructed memory on the environment (from ongoing interaction with the environment).

Each constructed memory is grounded as experience by its later use. As reinforced memories, experiences have the information of “in which situation, which action takes place and have which performance” and play a critical role in affecting new memory construction. Constructive memory enables agents to develop their experiences in a situated way. This allows the agent to reinterpret and learn new situations based on the augmented experience.

3. The Framework for a Situated Agent

3.1. PROPOSED FRAMEWORK

In this section, we describe a conceptual framework for a situated agent which is able to gain experiences and adapt itself to its use. The agent constructs its “experiences” from its internal representation processes (sensation, perception, conception and hypothesizing, expectation generation and modification), memory construction and grounding mechanisms. The adaptive tool changes its behaviours, which are classified as “reflexive”, “reactive” and “reflective” behaviour (Maher and Gero, 2002) depending on its reasoning, to adapt to its use. Here adaptation results from the totality of

the coordination of these behaviours within the process of memory construction and grounding. The framework of this situated agent is illustrated in Figure 2.

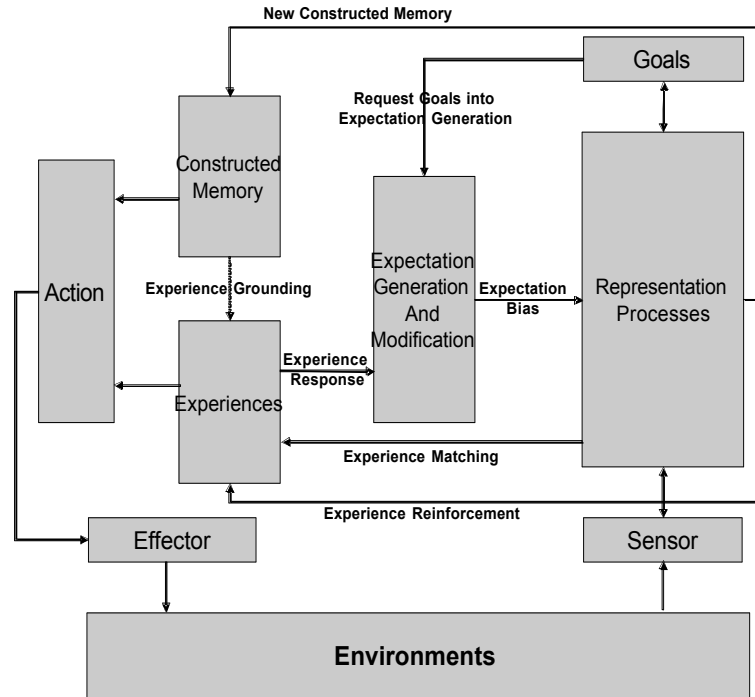


Figure 2. The conceptual framework for a situated agent.

3.2. AGENT REPRESENTATION PROCESSES

The internal representation processes consist of sensation, perception, conception, hypothesizing, expectation generation and modification. Each couples with one another in synthesizing data driven from the environment with expectation-driven experiences in order to form a constructed memory.

3.2.1. Sensation

Sensation is the process of generating sensory data from outside stimuli for further processing. During sensation, what an agent senses is biased by a number of issues: similar situations that occurred previously, what the current situation is, what the current goals are (goals are abstractions of expected situations associated with experiences which are constructed and grounded over time), and what the other processes involved are. Sensation is a holistic process of interaction with the environment during which new

situations are constructed involving external stimuli and internal coordination of past experiences, and goals.

3.2.2. Perception

Perception is the process of generating percepts from sequencing and coupling sensory data. Perceiving is more than mapping sensory stimuli to predefined category descriptions and can generate new information from transformation of stimuli sensed (Clancey, 1997). Perception also structures these adapted sensory data into sequence or simultaneous chunks (percepts) based on past sequences, coupled categorizations (perceptual experiences) and activated abstractions of percepts (concepts).

3.2.3. Conception

Conception is the process of categorizing perceptual sequences and chunks in order to form concepts. Concepts are abstractions of experience that confer a predictive ability for new situations (Rosenstein and Cohen 1998; Smith and Gero, 2000). Concepts give meaning to percepts which may be structured sequences or chunks of information.

3.2.4. Expectation Generation and Modification

Expectation generation and modification is the process of generating and modifying expected sensory data, percepts and concepts. Expectation is generated from matching the experiential response of the current situation with the agent's current goals. Expectation is related to the agent's view of possible consequences from its actions and affects its decision making. When an unexpected situation is recognized, it needs to be reinterpreted (Gero and Fujii, 2000). Reinterpretation occurs during the hypothesizing process in which focused concepts are selected for expectation generation and the causalities of failures are located in order to modify the expectations.

3.2.5. Hypothesizing

The hypothesizing process analyzes the possible causality of expectation failures and hypothesizes possible solutions. The trigger of hypothesizing is the mismatch between changes in the environment and the agent's expectations about the changes. It is where reinterpretation takes place in allowing the agent to learn from failures. Learning from failures is one of the major ways for an agent to improve its performance in uncertain environments. A situated agent reinterprets the design environment based on expectations which are regenerated from matching refocused concepts with the current situation. In order to produce a reasonable hypothesis, the agent may need to communicate with other agents for advice.

3.2.6. *Action*

Action is the process where decisions about changes to the environment are embodied. Agent's actions are effected as the design tool's behaviours. It is through action that agent's constructed memory is connected with the environment such that feedbacks from the environment can serve as cues of adjustment of the agent's behaviours.

3.2.7. *Sensor and Effector*

Sensor is the means by which the agent receives and gathers stimuli from environments. The agent gains access to the environment through sensor and affects the environment through its effector. The effector is the means by which the agent changes the environment through its actions.

3.3. EXPECTED ADAPTIVE BEHAVIOURS

Adaptation refers to the adjustment of behaviors to attain goals in the face of changes in the environment or the system itself (Kim, 1990). Here it is the agent's ability to change the tool's behaviours to achieve goals. The expected behaviours are defined as "reflexive behaviour", "reactive behaviour" and "reflective behaviour" (Maher and Gero, 2002).

3.3.1. *Reflexive Behaviour*

Reflexive behaviour occurs within the sensor – sensation – sensory experience – action – effector – environment loop, when the response to the current sensed data is sufficiently strong to produce a direct action without reasoning.

3.3.2. *Reactive Behaviour*

Reactive behaviour involves sensation, and perception in biasing the action process. The agent reacts to its environment, when the response to its current sensed and perceived data is such that the perceptual experiences in terms of habitual sequences or coupled information can produce actions. In its reactive behaviour, the agent matches its expectations with current percepts. Corresponding to the levels of similarity of the situation, it can reactivate perceptual experiences or construct a new memory. When the agent fails to coordinate its expectation with active changes in the environments, it cannot use reflexive or reactive reasoning.

3.3.3. *Reflective Behaviour*

Reflective behaviour is activated by discrepancies between expected percepts and current percepts – the failure of reactive behaviour. In its reflective behaviour, the agent coordinates all the processes. Sensation, and perception work along with conception, expectation generation and

modification, and hypothesizing processes in constructing the agent's expectation that represents the agent's belief about the possible incoming events and the consequences of the resulting actions.

In adapting to changes in the environment, the agent first activates sensation and perception and constructs expected sensory data and percepts from experiential responses. If the agent is unable to "comprehend" the situation based on its low level experiences (sensory, perceptual experiences), it modifies its behaviour to reflection. The agent activates the conception process in which current concepts and all its derivatives are generated from current percepts. The hypothesizing process locates focused concepts from current concepts, all derivatives of current concepts, current and expected percepts, and active goals. Focused concepts, as active abstractions of current percepts, need to be matched with conceptual experiences and re-generate expected percepts based on experiential responses. This is called "changing focuses of attention". If newly constructed expected percepts fit with the current situation, the agent then constructs a new memory from synthesizing expected percepts, current percepts and current focused concepts. Then this recently formed memory is used to produce actions to affect the environment through the agent's effector.

Upon receiving positive feedback from the environment (in terms of expected percepts by the user or other agents), the agent grounds the constructed memory into its experiences.

3.4. AGENT-ENVIRONMENT INTERACTION

Agents learn from their interactions with environments. This learning contributes to the competence of the agent, which is the way the agent acquires knowledge it requires to choose when and how to assist users (Maes, 1994). Among other usability challenges, such as trust, controllability, unobtrusiveness, privacy and breadth of experiences (Jameson, 2003), we are concerned here with obtrusiveness in a user-centered design process. Obtrusiveness refers to the extent to which the agent places demands on the user's attention which reduce user's ability to concentrate on their primary tasks (Jameson, 2003). We need to decide the approach for the agent to interact with user in a natural way – the right interaction at the right time.

In this section, we present an agent wrapper, as a member of a society of agents (Maher and Gero, 2002), that deals with interactions with design environments that include designer, the design and other agents in the society (Figure 1, Figure 3). Having such a wrapper learning and constructing memory from these interactions, the tool is able to change its behaviours and decide when and how to interact with user concerning what.

3.4.1. Agent-Environment Interaction

An agent's interactions with the user, the design and other agents enable the design tool to adapt its behaviours to its use, which as a result assists the usage of tool in design. What and how an agent learns becomes the fundamental question. We claim that an agent can learn from the usage of tool. The agent can learn from the user's habits and profiles of using tools. It can also construct experiences from the previous design problems. The agent that is situated in a society of agents also learns from other agents and cooperates with them in its reflective behaviour. Figure 3 depicts the above-mentioned agent-environment interaction that is a development of Figure 1. The designer interacts with the design tool in generating design artifacts. What the agent can do is to observe the designer's interaction with design tool (the way designer using tool), the design representations generated and the tool's states, from which the agent represents and constructs situational memory. The agent changes the tool's behaviours based on its reasoning about what is learned from these interactions and what was experienced previously. The agent also responds to the designer's activations and feedback.

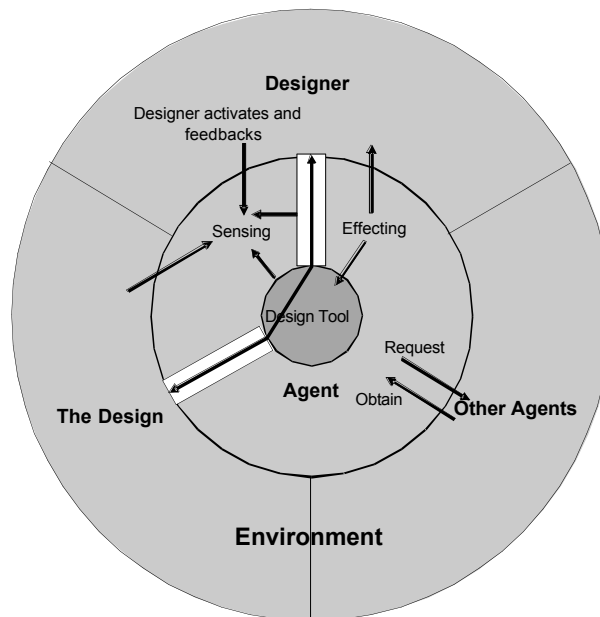


Figure 3. An illustration of agent-environment interaction.

3.4.2. Exemplar

Take as a simple exemplar, SwiftFile (Segal and Kephart, 1999; 2000) which is described as an intelligent assistant for Lotus Notes that helps users organize their e-mail into folders. It uses a text classifier to learn each user's mail-filing habits and predicts the three folders in which the user is most likely to place each incoming message, Figures 4 and 5.



Figure 4. Dialog box for filing messages in Lotus Notes. Typical users must select an appropriate folder from among dozens or hundreds of different choices (adapted from Segal and Kephart, 1999).

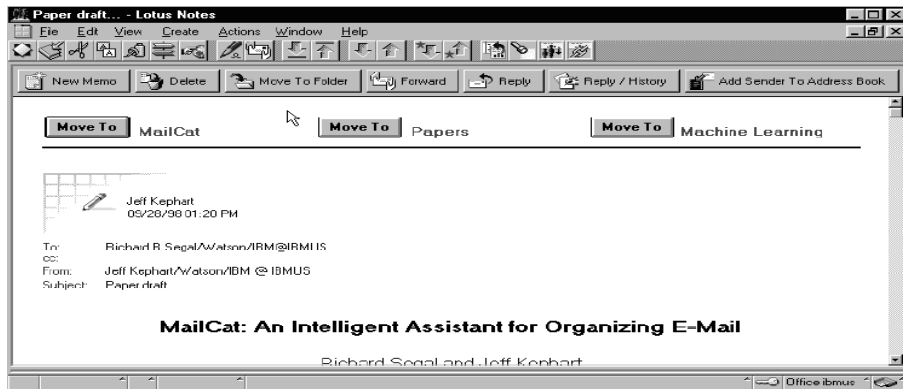


Figure 5. A screenshot of SwiftFile (adapted from Segal and Kephart, 1999).

The agent employed by SwiftFile can be viewed as a reactive agent since it simply applies machine learning algorithm (classification) to address users' repetitive behaviours. When the user's behaviour changes beyond the agent's understanding, it is not likely to adapt its behaviour to this change. Both the agent's learning ability and interactions are limited. In our framework, we endow the agent with learning capacities by which the agent

can represent changing situations and construct memory from experiential responses to these changes. Extending the agent-environment interactions to include interactions with the user, the design representation and other agents creates a richer environment for the agent to learn from.

In this part, we use Swiftfile as a vehicle to convey several scenarios of this situated agent-based assistance. We assume that Swiftfile adopts this situated agent-based assistant. As the agent senses the sequences of events of: “open mail files click”, “move to folder button”, “select folder click” and “move button”, it constructs a percept about a user’s profile from its interaction with environments, ie, “user AAA move mail files aaa to some folders”. The agent subsequently matches this percept with perceptual experiences for similar files movements in the past (classification may be applied in the matching process). The expected percepts are generated from synthesizing current percepts with matched experiences and displayed as a number of short-cut buttons – “mail file aaa is expected to move to bbb, ccc, or ddd folders”. If the expected percepts receive positive feedback from the user’s behaviour, the reactivated perceptual experiences are further reinforced. However, if the user changes behaviour from “move file” to “save attached files”, the agent experiences negative feedback which could be no clicks for short-cut buttons for a period of times. By doing so, the agent thus changes the tool’s behaviour to reflective behaviour in which the agent pulls more information from its internal representation processes and refocuses its attention to form new focused concept of the changed profile of the user. Based on the experiential response to the updated profile, the agent affects the use of the tool through revised shortcut buttons which represent agent’s expectation about user’s future behaviours.

The agent also interacts with the design in which the agent may have other experiences than the user in dealing with design problems. For instance, WEKA¹ is a suite of machine learning algorithms that can be embedded in this agent-based assistant to provide a matching process. The agent learns the patterns of the design problems through a variety of machine learning approaches offered by WEKA, i.e., classification, clustering and associative rules, etc. The processed design problems can then be compared with what was experienced before.

In an agent’s interactions in a society of agents the agent can learn from other agents which are more knowledgeable in certain situations. In its reflective behaviour, the agent collaborates with other agents to handle unfamiliar situations or consults with peers for knowledge that contains both solutions and recommendations for a suitable agent (Lashkari et al., 1994).

¹ <http://www.cs.waikato.ac.nz/ml/weka/>

4. Conclusion

This paper has presented and described an agent-based design assistant and its framework to allow a design tool to learn from and construct its experiences of usage from its interactions with the designer, the design and a society of agents. We also briefly mentioned the assistances and challenges posed by this approach. Based on both direct and indirect interactions with its environment, this agent wrapper enables us to build design tools that can adapt to their use, whilst, maintaining their objective knowledge. Additional work will further explore the interactions within a society of agents and will implement the proposed agent-based assistant and then test the benefits claimed.

Acknowledgements

This work is supported by a University of Sydney Sesqui R and D grant and by a Cooperative Research Centre for Construction Innovation (CRC-CI) Scholarship. The research is carried out at Key Centre of Design Computing and Cognition, University of Sydney, Australia.

References

- Clancey, W.: 1997, *Situated Cognition*, Cambridge University Press, Cambridge.
- Gero, J. S.: 1996, Design tools that learn: A possible CAD future, in B. Kumar (eds), *Information Processing in Civil and Structural Design*, Civil-Comp Press, Edinburgh, pp. 17-22.
- Gero, JS: 1998a, Conceptual designing as a sequence of situated acts, in I Smith (eds), *Artificial Intelligence in Structural Engineering*, Springer, Berlin, pp. 165-177.
- Gero, J. S.: 1998b, Towards a model of designing which includes its situatedness, in H. Grabowski, S. Rude and G. Grein (eds), *Universal Design Theory*, Shaker Verlag, Aachen, pp. 47-56.
- Gero, J. S.: 1999, Constructive memory in design thinking, in G. Goldschmidt and W. Porter (eds), *Design Thinking Research Symposium: Design Representation*, MIT, Cambridge, pp. 129-35.
- Gero, J. S.: 2003, Design tools as situated agents that adapt to their use, in W. Dokonal and U. Hirschberg (eds), *eCAADe21*, eCAADe, Graz University of Technology, pp. 177-180.
- Gero, J. S. and Fujii, H.: 2000, A computational framework for concept formation in a situated design agent, *Knowledge-Based Systems*, **13**(6): 361-368.
- Jameson, A.: 2003, Adaptive interfaces and agents, in J. Jacko and A. Sears (eds), *Human-computer interaction handbook*, Mahwah, NJ: Erlbaum, pp. 305-330.
- Kim, S. H.: 1990, *Design Intelligence: A Framework for Smart Systems*, Oxford University Press, New York.
- Lashkari, Y., Metral, M., and Maes, P.: 1994, Collaborative interface agents, *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press, pp. 444-449.
- Lieberman, H.: 2001, Introduction, in H. Lieberman (eds) *Your Wish is My Command: Programming by Example*, Morgan Kaufmann, San Francisco, pp. 1-7.

- Maes, P.: 1994, Agents that reduce work and information overload, *Communications of the ACM*, **37**: 31-40.
- Maher, ML and Gero, JS: 2002, Agent models of 3D virtual worlds, *ACADIA 2002: Thresholds*, California State Polytechnic University, Pomona, pp. 127-138.
- Malone, T., Grant, K., Turbak, F., Brobst, S. and Cohen, M.: 1987, Intelligent information-sharing systems, *Communications of ACM*, **30**(5): 390-402.
- Rosenstein, M. T. and Cohen, P. R.: 1998, Concepts from time series, *Proceedings of Fifteenth National Conference on Artificial Intelligence*, AAAI, Menlo Park, CA, pp. 739-745.
- Schon, D. A.: 1983, *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York.
- Segal, R. and Kephart, J.: 1999, MailCat: an intelligent assistant for organizing e-mail, *Proceedings of the Third International Conference on Autonomous Agents*, ACM Press, NY, pp. 276-282.
- Segal, R. and Kephart, J.: 2000, Incremental learning in SwiftFile, *Proceedings of the Seventh International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, pp. 863-870.
- Smith, G. and Gero, J. S.: 2000, The autonomous, rational design agent, in H. Fujii (eds) *Workshop on Situatedness in Design*, Artificial Intelligence in Design'00, Worcester, MA, pp. 19-23.
- Wooldridge, M. and Jennings, N. R.: 1995, Intelligent agents: Theory and practice, *Knowledge Engineering Review*, **10**(2): 115-152.

This is a copy of: Gero, JS and Peng, W (2004) A situated agent-based design assistant, in HS Lee and JW Choi (eds), *CAADRIA 2004*, Yonsei University Press, pp. 145-157.